

# Lab Assignment & Solution



Cybersecurity Professional Program  
Introduction to Python  
for Security

## File System & Error Handling

**PY-04-LS3**

**Handling Files**

**Note:** Solutions for the instructor are shown inside the green box.

## Lab Objective

Understand how to perform file operations while handling errors in Python.

## Lab Mission

Practice working with files and error handling.

## Lab Duration

10–20 minutes

## Requirements

- Basic knowledge of Python
- Working knowledge of an IDE environment
- Working knowledge of loops and error handling

## Resources

- Environment & Tools
  - Windows
    - PyCharm
    - Python 3

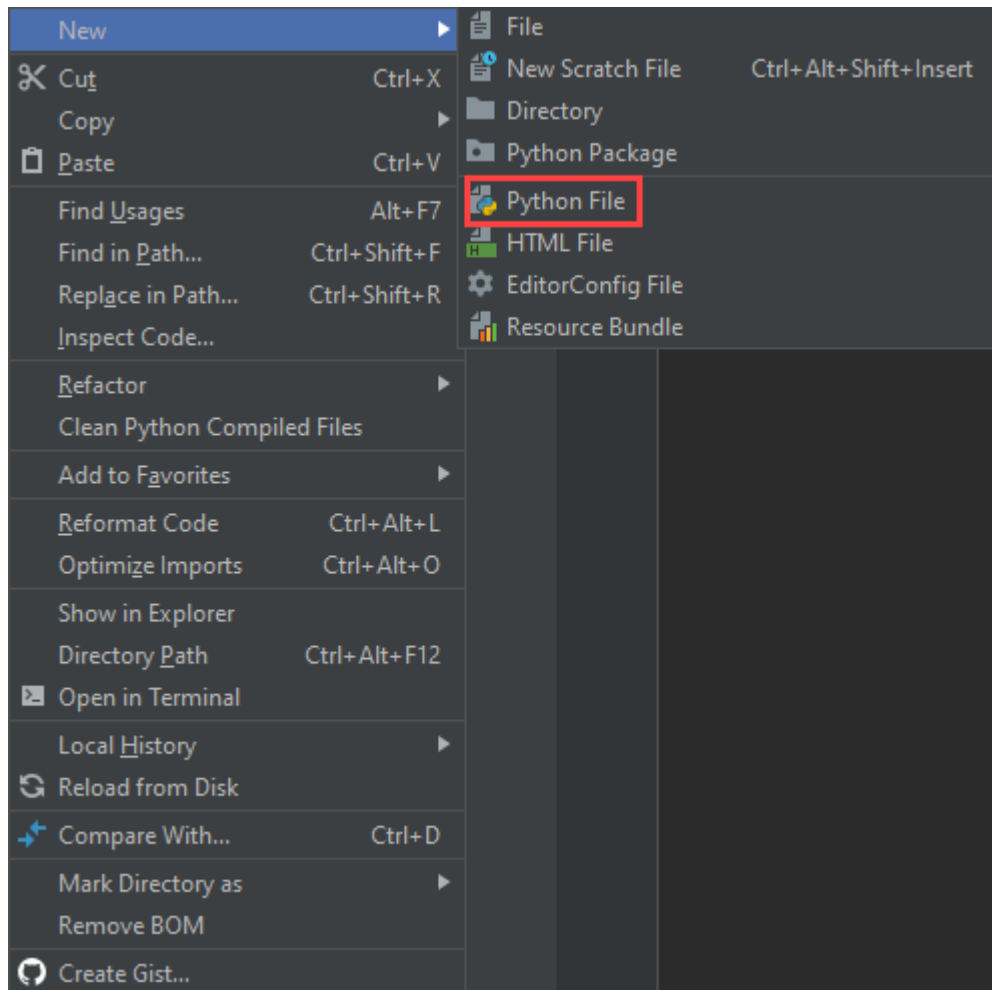
## Textbook References

- Chapter 4: File System and Error Handling
  - Section 2: File Manipulation

## Lab Task 1: Write Error

Create a ***with()*** statement that opens a file in read mode and then try to write to it.

- 1 Create a new Python file in PyCharm by right-clicking the project you created and selecting **New > Python File**.



- 2 Create a ***with()*** statement to open a text file in read mode with a variable.

```
with open("text.txt", "r") as text:
```

- 3 Write any text in the opened file.

```
    text.write("Test")
```

- 4 Put the code in a **try** block.

```
try:
    with open("text.txt", "r") as text:
        text.write("Test")
```

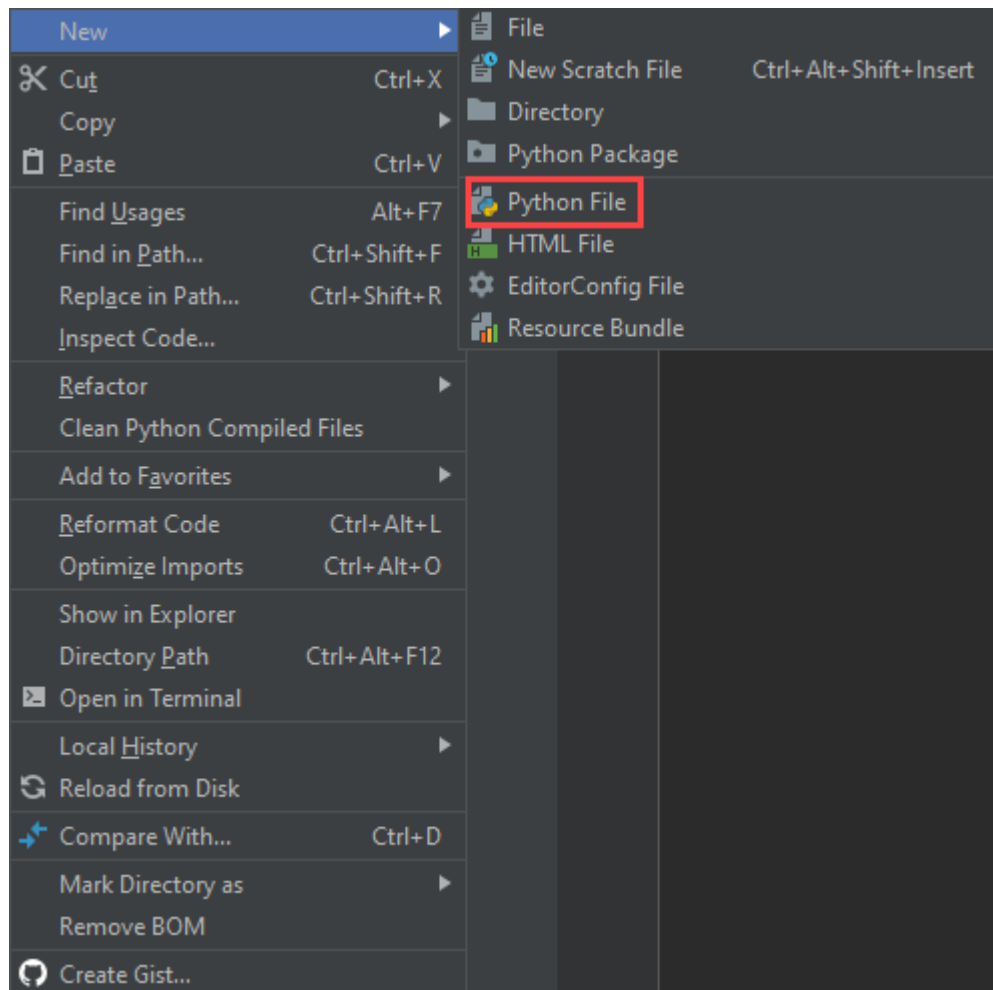
- 5 Create an exception block to handle the error and print an error message explaining that an open file cannot be written to in read mode.

```
try:
    with open("text.txt", "r") as text:
        text.write("Test")
except Exception:
    print("Unsupported Operation, cannot write in read mode.")
```

## Lab Task 2: Handling Files

Using the appropriate methods in Python, write a script that will open a file, take input from a user, and use the input to write to a file before exiting.

- 1 Create a new Python file in PyCharm by right-clicking the project you created and selecting **New > Python File**.



- 2 Open a non-existent file in append mode and set it to a variable. Note that using append to open a file will create a file if it doesn't exist.

```
file = open("file.txt", "a")
```

- 3 Add an infinite **while** loop to take user input and set it to a variable.

```
file = open("file.txt", "a")
while True:
    message = input("Enter text! ('Exit' to exit): ")
```

- 4 If the user input is *exit*, the loop will end.

**Note:** Make sure the user input is in lowercase letters.

```
file = open("file.txt", "a")
while True:
    message = input("Enter text! ('Exit' to exit): ")
    if message.lower() == "exit":
        break
```

- 5 If the input is not *exit*, write the input to the file and move to the next line.

```
file = open("file.txt", "a")
while True:
    message = input("Enter text! ('Exit' to exit): ")
    if message.lower() == "exit":
        break
    else:
        file.write(message + "\n")
```

- 6 Once the loop is done running, use the **close()** method to close the file.

```
file = open("file.txt", "a")
while True:
    message = input("Enter text! ('Exit' to exit): ")
    if message.lower() == "exit":
        break
    else:
        file.write(message + "\n")
file.close()
```

**7** Put your code in a **try** block.

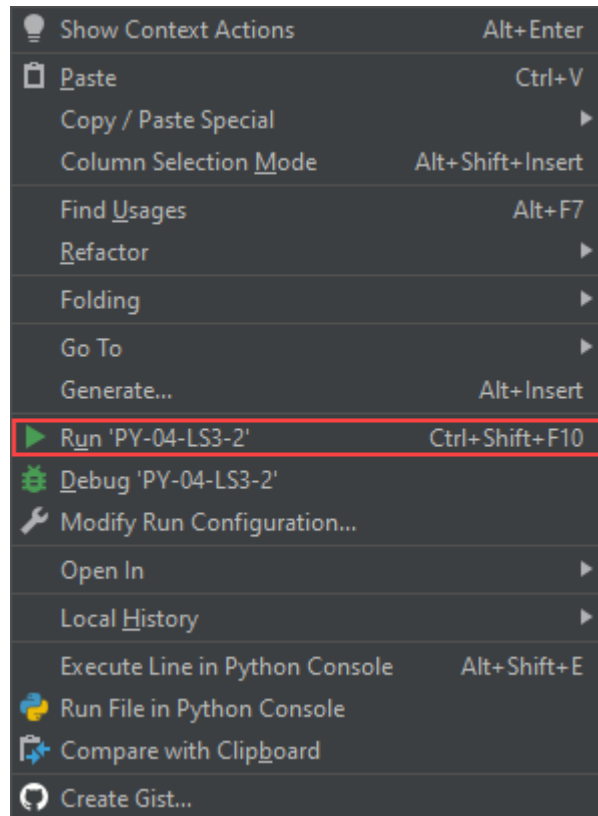
```
try:
    file = open("file.txt", "a")
    while True:
        message = input("Enter text! ('Exit' to exit): ")
        if message.lower() == "exit":
            break
        else:
            file.write(message + "\n")
    file.close()
```

**8** Add an **except** block after the **try** block to handle any unexpected errors while opening the file.

```
try:
    file = open("file.txt", "a")
    while True:
        message = input("Enter text! ('Exit' to exit): ")
        if message.lower() == "exit":
            break
        else:
            file.write(message + "\n")
    file.close()

except:
    print("An error occurred while trying to open the file.")
```

- 9 Run the code by right-clicking it and selecting **Run '[Python file name]'**.





- 10 Input several lines of text and exit the script. In the Project Files list in PyCharm, find the newly created **file.txt**, open it, and confirm that the text input during the script's execution is present in the file.

